UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/043,496	01/10/2002	Binyu Zang	42390P13145	9184

8791 7590 01/04/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
CHOW, CHIH CHING	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 01/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/043,496	ZANG ET AL.	
	Examiner	Art Unit	
	Chih-Ching Chow	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 February 2005* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *04/15/02*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to amendment dated September 26, 2005.

2. Per Applicants' request, claims 1, 11, and 21 have been amended.

3. Claims 1-30 remain pending.

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 09/02/2005 has been entered.

Drawings

5. FIG. 5 is objected, new corrected drawings in compliance with 37 CFR 1.121(d) are required in this application because FIG. 5, item 520 does not show the checking if there is another target register Rs, see Specification page 12, paragraph 0049, 2nd line. Drawing should be corrected. The requirement for corrected drawings will not be held in abeyance.

Specification

6. The ABSTRACT of the disclosure is objected to because the first line "a first last use of " should be "a last use of", see Specification paragraph 0014, line 3. Correction is required. See MPEP § 608.01(b).

7. 35 U.S.C. 112, first paragraph, requires the specification to be written in "full, clear, concise, and exact terms." The specification is replete with terms which are not clear, concise and exact. The specification should be revised carefully in order to comply with 35 U.S.C. 112, first paragraph. Examples of some unclear, inexact or verbose terms

used in the specification are: Many occurrences of "C1" should really be – Ci --, e.g. on page 10, paragraph 0034, 2nd line, after paragraph 0035, item iii, on page 12, paragraphs 0044 and 0047... etc., these mistakes have caused confusion of the Specification, the example is given on pages 10-12 are therefore hard to understand. A thorough review of the register names and appropriate corrections should be done.

8. The disclosure is objected to because of the following informalities: page 12, paragraph 0045, last line, the "rollback to original register r," should be "recovery", see FIG. 3, item 350. Appropriate correction is required.

Response to Arguments

9. Applicant's arguments with respect to claims 1, 11, and 21 have been considered but are moot in view of the new ground(s) of rejection.

10. The Examiner is maintaining the 35 USC § 112 (2) and the 35 USC § 103 Rejections. For the Applicants' convenience see 35 USC § 103 rejections (claims include the amendments) herein below:

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,631,514 by Bich-Cau Le (hereinafter "Le"), in view of US Patent No. 6,031,992 by Robert F. Cmelik et al., hereinafter "Cmelik" .

CLAIM	Le / Cmelik
1. A method comprising: determining based on a binary translation process a last use of a first canonical register in a block of code after a renaming, the first canonical register being mapped to a first original register; and applying based on a binary translation process one of a first rollback and a first recovery to the first original register based on whether the determined last use of the first canonical register occurs before a last definite write to the first original register in the block of code.	Le teaches a method based on binary translation process for register renaming, see Le's column 1, lines 32-37, "a **binary translator** permits programs or applications that were compiled for a **pre-existing architecture** (*original processor*) to be run on a new architecture (*target processor*) without having to recompile those applications. The **binary translator** translates the complied application into binary form which is used on the new system." Le's disclosure processes a block of code at a time, see column 6, lines 12-14, "Once the optimized code shown in FIG. 2C has been obtained, and executed, **if the execution is successful,** then the **next section of original code** is emulated." See Le's column 5, lines 61-64, "The translator would use **register renaming** to break all write-after-read (WAR) and write-after-write (WAW) register dependencies, allowing loads and most other operations to be scheduled speculatively." And column 6, lines 1-7, "During the renaming phase, a register map (RMAP) defining the dynamic legacy-to-native register mappings is computed. The initial RMAP maps each **legacy register** (*canonical register*) name R.sub.x to its corresponding **physical architectural register** *(original register)* r.sub.x. Every time a destination architectural **register is renamed,** the corresponding RMAP entry is set to the virtual register name, and a snapshot of the current RMAP is saved and associated with the instruction."—determining the last use or the last definite write can be done by checking the register map. Le teaches all aspects of claim 1 but does not mention the 'rollback' and 'recovery' specifically.

However, Cmelik teaches these features in an analogous art. For the rollback feature, see Cmelik, column 13, lines 2-8, "The target (or shadow) registers are connected to their working register equivalents through a dedicated interface that allows an operation called 'commit' to quickly transfer the content of all working registers to official target registers and allows an operation called '**rollback**' to **quickly transfer the content of all official target registers back to their working register** equivalents." For the recovery feature, see Cmelik, column 14, lines 24-28, "For the case where exceptions are used to detect failure of other speculations, such as whether an operation affects memory or memory mapped I/O, **recovery** is accomplished by the generation of new translations with different memory operations and different optimizations." It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Le's disclosure of the binary translation using register renaming method with 'rollback' and 'recover' concepts taught by Cmelik, for the purpose of efficiently using a large plurality of additional processor registers via register renaming to lesson the problem of instructions needing the same hardware resources. (see Cmelik, column 12, lines 61-65).

2. The method of claim 1 wherein applying one of the first rollback and the first recovery comprises:

applying the first rollback to the first original register if the determined last use of the first canonical register occurs before the last definite write to the first original

See claim 1 rejection, for the rollback part, see Cmelik column 18, lines 7-16, "If a target exception is generated during the running of any translated instruction or instructions, that exception is detected by the morph host hardware or software. In response to the detection of the target

register; and applying the first recovery to the first original register if the determined last use of the first canonical register does not occur before the last definite write to the first original register.	exception, the code morphing software 11 may cause the values (*if it is determined need to retain the value of the last use of the first canonical register before a last definite write occurs because of the execution is not successful*) retained in the official registers to be placed back into the working registers 41 and any non-committed memory stores in the gated store buffer 50 to be dumped (an operation referred to as "**rollback**")."
3. The method of claim 2 wherein applying the first rollback comprises: (a) replacing a first reference to a first target register with the first canonical register when the first reference is a destination of a last write to the first target register, the first target register corresponding to the first original register after the renaming; and (b) replacing a second reference to the first target register with the first canonical register when the second reference is a source of a first operation after the last write to the first target register.	For the feature of claim 2 see claim 2 rejection. For the rest of the feature in claim 3 see claim 1 rejection, here 'replacing' means 'writes to it' (write after read).
4. The method of claim 3 wherein applying the first recovery comprises: copying the first target register to the first canonical register at end of the block.	For the feature of claim 2 see claim 2 rejection. See claim 1 rejection, where quickly transfer the content (*copying*) of all official target registers back to their working register .
5. The method of claim 4 wherein copying the first target register comprises: (a) copying the first target register to a first unused temporary register; and (b) copying the first unused temporary register to the first canonical register.	For the feature of claim 4 see claim 4 rejection, for the rest of the feature of claim 5 see claim 4 rejection.
6. The method of claim 5 further	For the feature of claim 5 see claim 5

comprises:
recording a last use of a second canonical register in the block of code after the renaming, the second canonical register being mapped to a second original register; and
applying one of a second rollback and a second recovery to the second original register based on whether the determined second last use of the second canonical register occurs before a last definite write to the second original register in the block of code.

rejection. For the rest of the features see claim 1 rejection.

7. The method of claim 6 wherein applying one of the second rollback and the second recovery comprises:
applying the second rollback to the second original register if the determined last use of the second canonical register occurs before the last definite write to the second original register; and
applying the second recovery to the second original register if the determined last use of the second canonical register does not occur before the last definite write to the second original register.

For the feature of claim 6 see claim 6 rejection. For the rest of the features see claim 2 rejection.

8. The method of claim 7 wherein applying the second rollback comprises:
replacing a third reference to a second target register with the second canonical register when the third reference is a destination of a last write to the second target register, the second target register corresponding to the second original register after the renaming; and
replacing a fourth reference to the second target register with the second canonical register when the fourth reference is a source of a second operation after the last

For the feature of claim 7 see claim 7 rejection. For the rest of the features see claim 3 rejection.

write to the second target register.

9. The method of claim 8 wherein applying the second recovery comprises: copying the second target register to the second canonical register at end of the block.	For the feature of claim 8 see claim 8 rejection. For the rest of the features see claim 4 rejection.
10. The method of claim 9 wherein copying the second target register comprises: copying the second target register to a second unused temporary register before copying the first unused temporary register to the first canonical register; and copying the second unused temporary register to the second canonical register.	For the feature of claim 9 see claim 9 rejection. For rest of the features see claim 5 rejection.
11. A computer program product comprising: a machine useable medium having program code embedded therein, the program code comprising: computer readable program code to determine based on a binary translation process a last use of a first canonical register in a block of code after a renaming, the first canonical register being mapped to a first original register; and computer readable program code to apply based on the binary translation process one of a first rollback and a first recovery to the first original register based on whether the determined last use of the first canonical register occurs before a last definite write to the first original register in the block of code.	Both Le's FIG. 1 and Cmelik's Fig. 9 teaches a machine useable medium having code embedded therein. For the rest of feature 11, same as claim 1 rejection.
12. The computer program product of claim 11 wherein the computer readable program code to apply one of the first rollback and the first recovery comprises:	For the feature of claim 11 see claim 11 rejection. For rest of the features see claim 2 rejection.

computer readable program code to apply the first rollback to the first original register if the determined last use of the first canonical register occurs before the last definite write to the first original register; and

computer readable program code to apply the first recovery to the first original register if the determined last use of the first canonical register does not occur before the last definite write to the first original register.

13. The computer program product of claim 12 wherein the computer readable program code to apply the first rollback comprises:

computer readable program code to replace a first reference to a first target register with the first canonical register when the first reference is a destination of a last write to the first target register, the first target register corresponding to the first original register after the renaming; and

computer readable program code to replace a second reference to the first target register with the first canonical register when the second reference is a source of a first operation after the last write to the first target register.

For the feature of claim 12 see claim 12 rejection. For rest of the features see claim 3 rejection.

14. The computer program product of claim 13 wherein the computer readable program code to apply the first recovery comprises:

computer readable program code to copy the first target register to the first canonical register at end of the block.

For the feature of claim 12 see claim 12 rejection. For rest of the features see claim 4 rejection.

15. The computer program product of claim 14 wherein the computer readable program code to copy the first target register comprises:

For the feature of claim 14 see claim 14 rejection. For rest of the features see claim 5 rejection.

computer readable program code to copy the first target register to a first unused temporary register; and

computer readable program code to copy the first unused temporary register to the first canonical register.

16. The computer program product of claim 15 further comprises:

computer readable program code to record a last use of a second canonical register in the block of code after the renaming, the second canonical register being mapped to a second original register; and

computer readable program code to apply one of a second rollback and a second recovery to the second original register based on whether the determined last use of the second canonical register occurs before a last definite write to the second original register in the block of code.

For the feature of claim 15 see claim 15 rejection. For rest of the features see claim 6 rejection.

17. The computer program product of claim 16 wherein the computer readable program code to apply one of the second rollback and the second recovery comprises:

computer readable program code to apply the second rollback to the second original register if the determined last use of the second canonical register occurs before the last definite write to the second original register; and

computer readable program code to apply the second recovery to the second original register if the determined last use of the second canonical register does not occur before the last definite write to the second original register.

For the feature of claim 16 see claim 16 rejection. For rest of the features see claim 7 rejection.

18. The computer program product of claim

For the feature of claim 17 see claim 17

17 wherein the computer readable program code to apply the second rollback comprises: computer readable program code to replace a third reference to a second target register with the second canonical register when the third reference is a destination of a last write to the second target register, the second target register corresponding to the second original register after the renaming; and computer readable program code to replace a fourth reference to the second target register with the second canonical register when the fourth reference is a source of a second operation after the last write to the second target register.	rejection. For rest of the features see claim 8 rejection.
19. The computer program product of claim 18 wherein the computer readable program code to apply the second recovery comprises: computer readable program code to copy the second target register to the second canonical register at end of the block.	For the feature of claim 18 see claim 18 rejection. For rest of the features see claim 9 rejection.
20. The computer program product of claim 19 wherein the computer readable program code to copy the second target register comprises: computer readable program code to copy the second target register to a second unused temporary register before copying the first unused temporary register to the first canonical register; and computer readable program code to copy the second unused temporary register to the second canonical register.	For the feature of claim 19 see claim 19 rejection. For rest of the features see claim 10 rejection.
21. A system comprising: a binary translation processor; and	Both Le's FIG. 1 and Cmelik's Fig. 9 teaches a system with a processor and

a memory coupled to the processor to store program code, the program code, when executed, causing the processor to:

determine a last use of a first canonical register in a block of code after a renaming, the first canonical register being mapped to a first original register; and

apply one of a first rollback and a first recovery to the first original register based on whether the determined last use of the first canonical register occurs before a last definite write to the first original register in the block of code.

memory to store program code. For the rest of feature 21, same as claim 1 rejection.

22. The system of claim 21 wherein the program code causing the processor to apply one of the first rollback and the first recovery causes the processor to:

apply the first rollback to the first original register if the determined last use of the first canonical register occurs before the last definite write to the first original register; and

apply the first recovery to the first original register if the determined last use of the first canonical register does not occur before the last definite write to the first original register.

For the feature of claim 21 see claim 21 rejection. For rest of the features see claim 2 rejection.

23. The system of claim 22 wherein the program code causing the processor to apply the first rollback causes the processor to:

replace a first reference to a first target register with the first canonical register when the first reference is a destination of a last write to the first target register, the first target register corresponding to the first original register after the renaming; and

replace a second reference to the first target register with the first canonical

For the feature of claim 22 see claim 22 rejection. For rest of the features see claim 3 rejection.

register when the second reference is a source of a first operation after the last write to the first target register.

24. The system of claim 23 wherein the program code causing the processor to apply the first recovery causes the processor to:
copy the first target register to the first canonical register at end of the block.

For the feature of claim 22 see claim 22 rejection. For rest of the features see claim 4 rejection.

25. The system of claim 24 wherein the program code causing the processor to copy the first target register causes the processor to:
copy the first target register to a first unused temporary register; and
copy the first unused temporary register to the first canonical register.

For the feature of claim 24 see claim 24 rejection. For rest of the features see claim 5 rejection.

26. The system of claim 25 wherein the program code further causes the Processor to:
determine a last use of a second canonical register in the block of code after the renaming, the second canonical register being mapped to a second original register; and
apply one of a second rollback and a second recovery to the second original register based on whether the determined last use of the second canonical register occurs before a last definite write to the second original register in the block of code.

For the feature of claim 25 see claim 25 rejection. For rest of the features see claim 6 rejection.

27. The system of claim 26 wherein the program code causing the processor to apply one of the second rollback and the second recovery causes the processor to:
apply the second rollback to the second

For the feature of claim 26 see claim 26 rejection. For rest of the features see claim 7 rejection.

original register if the determined last use of the second canonical register occurs before the last definite write to the second original register; and

apply the second recovery to the second original register if the determined last use of the second canonical regiester does not occur before the last definite write to the second original register.

28. The system of claim 27 wherein the program code causing the processor to apply the second rollback causes the processor to:

replace a third reference to a second target register with the second canonical register when the third reference is a destination of a last write to the second target register, the second target register corresponding to the second original register after the renaming; and

replace a fourth reference to the second target register with the second canonical register when the fourth reference is a source of a second operation after the last write to the second target register.

For the feature of claim 27 see claim 27 rejection. For rest of the features see claim 8 rejection.

29. The system of claim 28 wherein the program code causing the processor to apply the second recovery causes the processor to:

copy the second target register to the second canonical register at end of the block.

For the feature of claim 28 see claim 28 rejection. For rest of the features see claim 8 rejection.

30. The system of claim 29 wherein the program code causing the processor to copy the second target register causes the processor to:

copy the second target register to a second unused temporary register before

For the feature of claim 29 see claim 20 rejection. For rest of the features see claim 10 rejection.

copying the first unused temporary register
to the first canonical register; and
copy the second unused temporary
register to the second canonical register.

Conclusion

The following summarizes the status of the claims:

35 USC § 103 rejection: Claims 1-30

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 7:30am - 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature of relating to the status of this application should be directed to the **TC2100 Group receptionist: 571-272-2100**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow
Examiner
Art Unit 2192
December 14, 2005

TUAN DAM
SUPERVISORY PATENT EXAMINER

CC